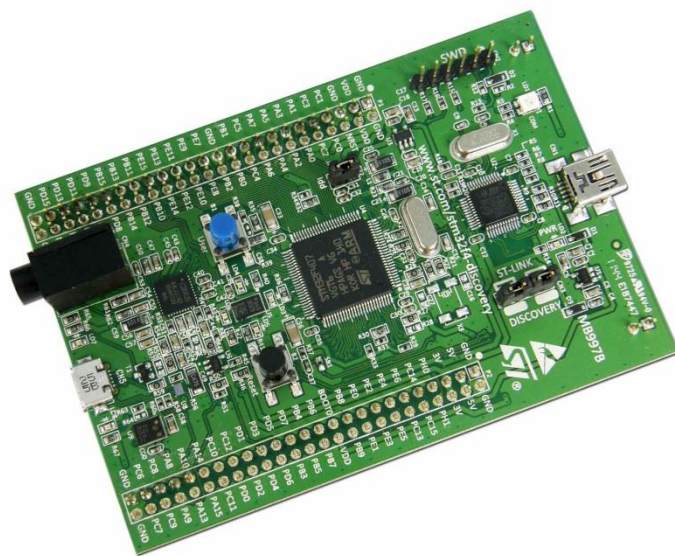


Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej

# Podstawy technik mikroprocesorowych

Ćwiczenia laboratoryjne SPI



Michał Fularz, Adam Owczarkowski  
2013-12-17

## 1. Wstęp teoretyczny.

Na zajęcia obowiązuje znajomość materiału zaprezentowanego na wykładzie.

Korzystając z wiadomości z wykładu i strony:

[http://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface\\_Bus](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)

odpowiedzieć na następujące pytania:

- jakie sygnały są wykorzystywane przez SPI?
- do czego służy pin CS?
- ile pinów GPIO jest niezbędnych do obsługi SPI przez mikrokontroler (master) z trzema urządzeniami (slave)?
- jakie są dostępne częstotliwości pracy SPI?
- ile danych jest przesyłanych w czasie jednego cyklu zegara?

## 2. Przebieg ćwiczenia.

### a. Tworzenie projektu.

Utworzyć nowy projekt (lub skorzystać z programu z poprzednich zajęć). W zakładce **Repositories** dodać następujące moduły:

- SPI – funkcje obsługi interfejsu (dodaje pliki `stm32f4xx_spi.h` oraz `stm32f4xx_spi.c`).

W oknie edytora kodu otworzyć załączone pliki (aby je podświetlić w oknie **Components** wybrać odpowiedni moduł (SPI). Zwrócić uwagę na:

- Exported types (`SPI_InitTypeDef`),
- Exported constants,
- Exported functions (`SPI_Init`, `SPI_I2S_SendData`, `SPI_I2S_ReceiveData`).

W pliku `stm32f4xx_spi.c` znajduje się krótki opis funkcjonalności modułu (na początku pliku, w komentarzu), a także definicje funkcji wraz z ich krótkim opisem (indeksowanym przez IDE i wyświetlanym po umieszczeniu kursora na funkcji i naciśnięciu klawisza F2).

Do pliku z funkcją **main** programu dodać odpowiednie biblioteki (`#include "stm32f4xx_spi.h"`).

## b. Konfiguracja interfejsu SPI.

Przed przystąpieniem do konfiguracji modułu SPI, konieczne jest skonfigurowanie i włączenie generatora sygnału zegarowego, a następnie doprowadzenia sygnału zegarowego do układów peryferyjnych, które mają być wykorzystane:

```
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE); // zegar
dla portu GPIO z którego wykorzystane zostaną piny do SPI (MOSI,
MISO, SCK)
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOE, ENABLE); // zegar
dla portu GPIO z którego wykorzystany zostanie pin do SPI (CS)
RCC_APB1PeriphClockCmd(RCC_APB1Periph_SPI2, ENABLE); // zegar
dla modułu SPI2
```

Domyślnie, po włączeniu zasilania, do bloków funkcjonalnych nie jest doprowadzony sygnał zegarowy, co umożliwia obniżenie poboru prądu.

Konfiguracji samych portów dokonuje się wykorzystując funkcję GPIO\_Init. Funkcja ta jako parametry przyjmuje nazwę portu, który ma zostać zainicjalizowany oraz strukturę typu GPIO\_InitTypeDef. Poszczególne ustawienia portu wprowadza się inicjalizując pola struktury.

```
GPIO_InitTypeDef GPIO_InitStructure;

//inicjalizacja pinów wykorzystywanych do SPI
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
// SPI SCK, MISO, MOSI
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13 | GPIO_Pin_14 |
GPIO_Pin_15;
GPIO_Init(GPIOB, &GPIO_InitStructure);

// konfiguracja pinu CS
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOE, &GPIO_InitStructure);
// deselecy - ustawienie pinu w stan wysoki
GPIO_SetBits(GPIOE, GPIO_Pin_3);
```

W przypadku realizacji przez wybrane piny funkcji dedykowanych (alternatywnych wobec GPIO, jak np. obsługa modułu SPI) niezbędna jest ich konfiguracja do tego celu:

```
GPIO_PinAFConfig(GPIOB, GPIO_PinSource13, GPIO_AF_SPI2); // SCK
GPIO_PinAFConfig(GPIOB, GPIO_PinSource14, GPIO_AF_SPI2); // MISO
GPIO_PinAFConfig(GPIOB, GPIO_PinSource15, GPIO_AF_SPI2); // MOSI
```

Aby skonfigurować moduł SPI należy, podobnie jak w przypadku portów wejścia/wyjścia, zainicjalizować pola dedykowanej dla tego układu peryferyjnego struktury.

### Konfiguracja modułu SPI w trybie MASTER:

```
// konfiguracja SPI w trybie MASTER
SPI_InitTypeDef SPI_InitStructure;
SPI_I2S_DeInit(SPI2);
SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
SPI_InitStructure.SPI_CPHA = SPI_CPHA_1Edge;
SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_4;
SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
SPI_InitStructure.SPI_CRCPolynomial = 7;
SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
SPI_Init(SPI2, &SPI_InitStructure);
```

### Konfiguracja modułu SPI w trybie SLAVE:

```
// konfiguracja SPI w trybie SLAVE
SPI_InitTypeDef SPI_InitStructure;
SPI_I2S_DeInit(SPI2);
SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
SPI_InitStructure.SPI_CPHA = SPI_CPHA_1Edge;
SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_4;
SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
SPI_InitStructure.SPI_CRCPolynomial = 7;
SPI_InitStructure.SPI_Mode = SPI_Mode_Slave;
SPI_Init(SPI2, &SPI_InitStructure);
```

### Można teraz uruchomić moduł SPI2:

```
SPI_Cmd(SPI2, ENABLE);
```

Wysyłanie danych przez układ MASTER odbywa się przy użyciu poniższego fragmentu kodu:

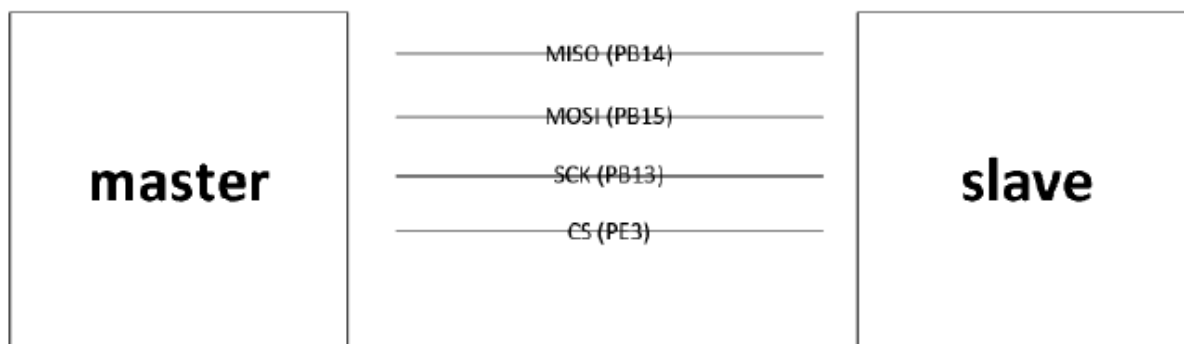
```
SPI_I2S_SendData(SPI2, 0x3);
while(SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE) == RESET);
```

Odbieranie danych przez układ SLAVE odbywa się przy użyciu poniższego fragmentu kodu:

```
while(SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_RXNE) == RESET);
Recevie[0] = SPI_I2S_ReceiveData(SPI2);
```

### Zadanie:

Korzystając z powyższych fragmentów kodu połączyć ze sobą dwie płytki według poniższego schematu. Przesyłać dane do układu SLAVE i wizualizować je na diodach (np. najmłodsze 4 bity). UWAGA w konfiguracji modułu ustawiono programowe sterowanie pinem CS (SPI\_NSS\_Soft), w związku z czym należy przed nadawaniem danych ustawić pin w odpowiedni stan.



### **c. Obsługa akcelerometru.**

Korzystając z biblioteki dostarczonej przez prowadzącego zrealizować podane na zajęciach zadania.